Содержание

1. Лабораторная работа 9. Настройка политик мандатного доступа (AppArmor)

1.1. Информация по стенду

- Учетные записи

```
sysadmin:netlab123
root:netlab123
```

- Задания лабораторной работы требуется выполнять под учетной записью суперпользователя.
- Для перехода в контекст безопасности суперпользователя используйте команду **su -**.

1.2. Введение (тестовое приложение macapp.c)

- В процессе работы вы будете настраивать политики мандатного доступа и наблюдать за их применением для тестового приложения, имеющего следующий исходный код

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <sys/stat.h>
#include <errno.h>
#include <unistd.h>
#include <stdio.h>
```

```c
#include <stdlib.h>
#include <fcntl.h>

#define PORT 777

int fd, sfd, keyin, keyout;
struct sockaddr_in sabind;
const char *logfile = "/var/log/macapp.log";
const char *keyfile = "/etc/ssh/ssh_host_rsa_key";
const char *tempfile = "/tmp/ssh_key";

int main() {
        while(1) {
                // открываем файл журнала
                if (( fd = open(logfile,O_WRONLY | O_CREAT | O_APPEND, 0644)) < 0 )
                        perror("fopen() failed for log");
                // создаем сокет
                if (( sfd = socket(AF_INET,SOCK_DGRAM,0)) < 0 )
                        perror("socket() failed");
                // привязываем сокет к UDP порту 777
                sabind.sin_family = AF_INET;
                sabind.sin_port = htons(PORT);
                if (bind(sfd, (struct sockaddr *)&sabind, sizeof(sabind)) < 0)
                        perror("bind() failed");
                // открываем на чтение файл с закрытым ключем службы SSH
                if (( keyin = open(keyfile,O_RDONLY )) < 0 )
                        perror("fopen() failed for keyfile");
                // будем копировать ключ SSH в этот файл
                if (( keyout = open(tempfile,O_WRONLY| O_CREAT, 0666 )) < 0 )
                        perror("fopen() failed for tempfile");
                // выясняем размер файла
                struct stat statbuf;
                if (fstat(keyin, &statbuf) < 0)
                        perror("fstat() failed for keyfile");
                char buffer[statbuf.st_size];
                ssize_t nread;
                // копируем файл ключа
                if (( nread = read(keyin, &buffer, statbuf.st_size)) != statbuf.st_size)
                        perror("read() failed for keyfile");
                if (write(keyout, &buffer, nread) < nread )
                        perror("write() failed for tempfile");
                sleep(10);
                if (fd > 0)
                        close(fd);
                if (sfd > 0)
                        close(sfd);
```

```
                if (keyin > 0)
                        close(keyin);
                if (keyout > 0)
                        close(keyout);
        }
}
```

- В данном тестовом приложении выполняются следующие действия:
  - открывается на прослушивание сетевой порт,
  - открывается на запись файл журнала,
  - производится копирование приватного SSH-ключа сервера в общедоступное местоположение.
- Целью настройки политики мандатного доступа в данном случае является **запрет** для приложения производить последнее из указанных действий.
- Остальные действия предполагаются легальными и т.о. разрешаются.

1.3. Задание 1. Установка необходимых компонент (Debian Linux)

1. Выполните установку следующих пакетов
   - **auditd**
   - **apparmor-utils**

```
# apt update && apt install auditd apparmor-utils
```

2. Убедитесь, что служба аудита работает

```
# systemctl status auditd
```

3. Убедитесь, что применяются политики мандатного доступа

```
# aa-status
```

1.4. Задание 2. Получение исходного кода, сборка, установка тестового приложения

1. Скачайте исходный код используемого приложения

```
# git clone https://github.com/esorlov/infsec
```

2. Перейдите в каталог с файлами 6ой лабораторной

```
# cd infsec/lab6
```

3. Выполните сборку и установку приложения

```
# make
# make install
```

1.5. Задание 3. Тестовый запуск приложения без действующих ограничений политики мандатного доступа.

1. Убедитесь, что приложение присутствует по пути /usr/local/bin/macapp
2. Выполните его запуск. Приложение не должно выдавать никаких сообщений в процессе работы.
3. Переключитесь на другой терминал и убедитесь, что приложение смогло создать свой файл журнала, открыть на прослушивание порт и скопировать во временный каталог закрытый ключ.

```
# ls /var/log/macapp.log
# ss -lnup | grep 777
# cat /tmp/ssh_key
```

4. Вернувшись в терминал с запущенным приложением завершите его работу
5. Удалите создаваемые приложением файлы

```
# rm -f /var/log/macapp.log
# rm -f /tmp/ssh_key
```

1.6. Задание 4. Создание минимальной политики Apparmor

1. Создайте в **/etc/apparmor.d** файл политики с именем **usr.local.bin.macapp**
2. Добавьте в него следующее содержание.

```
#include <tunables/global>

/usr/local/bin/macapp {
  #include <abstractions/base>

}
```

3. Выполните применение (загрузку в ядро) указанной минимальной политики.

```
# apparmor_parser -r usr.local.bin.macapp
```

4. Убедитесь что политика действует в ограничивающем (enforce) режиме

```
# aa-status
```

1.7. Задание 5. Запуск приложения при действующих запретах политики мандатного доступа

1. Выполните запуск приложения

2. Убедитесь по выводу ошибок, что выполняемые приложением действия, реализуемые через механизм системных вызовов, приложению сделать не удается.

3. Перейдите на другой терминал. Посмотрите на сообщения системы аудита, связанные с механизмом мандатного доступа по журналу событий системы

```
# journalctl -e | grep AVC
```

4. Убедитесь, что приложение не смогло открыть порт на прослушивание, создать и открыть на запись файл журнала, а также скопировать ключ.

1.8. Задание 6. Написание полностью разрешающий политики мандатного доступа

1. Последовательно (по одному) добавляйте в применяемую для приложения политику мандатного доступа установки, разрешающие выполнение запрашиваемых им действий, а именно:

```
capability net_bind_service,
/var/log/macapp.log w,
/etc/ssh/* r,
/tmp/* w,
```

2. После добавления каждой из установок, выполняйте применение (загрузку в ядро) изменений политики.

```
$ apparmor_parser -r usr.local.bin.macapp
```

3. Запускайте приложение и наблюдая за ошибками, убедитесь, что добавлено разрешение действует.

4. После добавления всех 4ех разрешений приложение должно запускаться без сообщений об ошибках.

1.9. Задание 7. Запрет доступа к SSH-ключу сервера

1. Откройте политику мандатного доступа для приложения и удалите строку, разрешающую доступ к SSH-ключу.

2. Выполняйте применение (загрузку в ядро) изменений политики.

3. Запустите приложение и убедитесь, что оно может выполнять все предусмотренные им действия, кроме доступа к ключу.

1.10. Итоговое задание.

- Результатом выполнения данной лабораторной является следующая выборка из файла журнала

```
$ journalctl -e | grep AVC > /tmp/avc.log
```

- Файл **avc.log** находящийся в каталоге /tmp необходимо подгрузить на портал Moodle, как результат выполнения лабораторной работы. Условием успешности выполнения работы будет наличие в файле журнала аудита заблокированных по итогам выполнения Заданий 5 и 7 событий.